Author Names & Affiliations

- Micah Beck - University of Tennessee
- Terry Moore - University of Tennessee

Contact Email Address (for NSF use only)

(Hidden)

Research Domain, discipline, and sub-discipline

Information Technology Infrastructure, Network Architecture, Distributed Cloud Service

Title of Submission

Creating A Globally Scalable, Converged ICT Infrastructure

Abstract (maximum ~200 words).

Historically, the Internet emerged as an evolution of circuit-based networking that unified telephony and data communication, exposing a common packet delivery abstraction at layer 3. This leaves abstractions of storage and processing out of the picture, providing no standard, interoperable way for Internet intermediate nodes to be used to implement generalized distributed services.

The conventional account of the network stack describes layer 2 as the "Link Layer", offering only the constituent communication services using which the Internet Protocol Suite is built at layer 3. In our model, the layer that is used to implement Layer 3 includes all ICT services including storage and processing. We consider this a model of layer 2 that generalizes Internet stack's Link Layer.

The Internet model places the burden of service creation on end nodes that use the network only for datagram delivery service. Lower layer services are not accessible to the implementation of those services through the standard layer 3 interface. The research challenge we pose is the creation of a global, converged (general) ICT infrastructure that exhibits deployment scalability characteristics that we associate with the Internet. Our proposed approach is to move the locus of interoperation to our generalized notion of layer 2.

Question 1 Research Challenge(s) (maximum ~1200 words): Describe current or emerging science or engineering research challenge(s), providing context in terms of recent research activities and standing questions in the field.

A fundamental problem facing nearly every field in the new era of data intensive science and engineering falls under the heading of "data logistics," i.e., the management of the time sensitive positioning and encoding/layout of data relative to its intended users and the resources they can use. At a minimum, all communities who want to analyze data that is generated at one (or a few) locations, but is worked on

somewhere else, confront challenges of data logistics in this sense. The challenges any general solution to the problem must face are easy to see: 1) the size of the individual data objects can be large, and certainly collections of them are often immense; 2) the users (both actual and potential) are highly distributed, geographically and socially, with widely varying degrees of access to suitable network bandwidth; 3) in some communities (e.g. remote sensing), new data are constantly flowing in, so that user and community collections frequently need to be updated from remote sources; 4) the data are often multidimensional and complex; 5) rounds of "second order" data distribution are often desired for highly valued data products generated by users at network's edge; 6) and finally, any proposed cyberinfrastructure solution must support the kind of interoperability and multidimensional scalability that is required for long-run sustainability.

Furthermore, it is widely agreed that this problem is going to get worse, not just for fields of research and engineering, but for economic, social, and military application areas as well. The authors of a recent NSF report on future directions in wireless networking observe that the pervasive deployments of sensor technologies (e.g. cameras, microphones, probes, etc.), which are fundamental to enthusiastic visions of "Smart Cities" and the Internet of Things (IoT), will "...generate massive data inflows [that produce] as much if not more data and network traffic than the World Wide Web," and will therefore "... reverse current loads, where most data is produced in the cloud and consumed at the edge." This remarkable reversal of direction of the data tide, which turns the familiar "last mile problem" into a multidimensional "first mile problem," will clearly demand revolutionary innovation in distributed computing systems in "edge environments," i.e. outside of the tightly coupled set of large data centers that constitute today's Cloud infrastructures. It seems clear that extremely high levels of parallelism and scalability (in multiple dimensions) will be required to meet all the challenges raised by this explosive growth in data production within edge environments. Contrary to the widespread opinion that these goals can be achieved by going further down the current architectural path, i.e., by expanding on current approaches to cloud computing, we only a novel and extreme approach to distributed computing that generalizes and integrates wide area networking, storage and processing (ICT) infrastructure will successfully meet the steep parallelism and scalability challenges of data-driven applications in edge environments.

The growing "data deluge" is finally adding a sense of urgency to a long-standing aspiration of the distributed computing community. Efforts to generalize and integrate wide area networking, storage and processing in ICT infrastructure by exposing a common model of the network node predate the global Internet. The advantages of creating a global, interoperable wide area infrastructure analogous to the one that historically supported telephony and the one that now supports the Internet are well understood and have been continuously researched, prototyped and deployed for testing over the past 40 years. The problem is not a lack of candidate architectures, ambition or political will. The challenge is to create an infrastructure that exhibits Internet-like deployment scalability.

By "deployment scalability" we mean the inherent ability of an infrastructure to grow beyond conventional boundaries in many different dimensions, including but not limited to number of constituent components (nodes in networking terms), across administrative domains, geographical obstacles, and including legal, cultural, and many others. We refer to this form of scalability as inherent or "by design" in order to contrast it with the kind of scalability that is imposed by legislation or monopolistic market control. The term "deployment" is intended to distinguish it from certain more restricted meanings of the term "scalability", such as per-application performance and number of components connected at a centralized location.

The list of successful ICT infrastructures that exhibit deployment scalability is short. Notable among them are the Internet itself and the Unix kernel interface (which is not a unified physical infrastructure but a software standard whose "users" are applications that are built on it). Both of these are examples of "hourglass" architectural design, which seeks to define a common interface that is a "thin waist", meaning that it is minimal and orthogonal. This style of interface design, described by Saltzer, Clark and Reed in their classic paper, can be paraphrased as being "simple, generic and limited".

The definition of thin-waisted designs in ICT infrastructure, their promulgation and possible eventual widespread acceptance has been difficult. This is particularly true when an entrenched standard interface at one layer is supplanted by a thin interface at a lower layer. From a technical point of view, some features of the higher layer service that are viewed as necessary by important stakeholders may not be easily supported by the new thin standard, especially not at first. The trade-off offered is usually greater generality, with a new standard at the lower layer supporting alternatives to the legacy higher-layer standard. The creation of competing alternatives may not be welcomed by stakeholders whose personal or corporate strategies have been predicated on exclusivity of the legacy standard.

Some aspects of the legacy high level standard may be viewed as definitional to the services offered at that level, to the point that weakening them may be seen as making the infrastructure unusable or unmanageable. This form of rejectionism may be motivated by entrenched interests, as described above, but it may also be the result of entrenched habits of thought. Categories of technology, industrial categories, and entire fields of academic study may be defined in terms of high layer abstractions and services which have been useful tools but also limit freedom of design and constrain innovation.

In digital telephony, per-use monitoring and billing of resource utilization was fundamental to the business model and the ability to

implement it accurately was an unquestioned constraint on the entire infrastructure. Quality of Service guarantees were another class of unquestioned features that consumed enormous intellectual and technological resources. The Internet has been successful using greatly weakened versions of these features. The approach to globally scalable, converged ICT infrastructure for the emerging era of data-intensive research and engineering aims to achieve the same kind of minimal sufficiency in the fundamental services at the base of its software stack.

Question 2 Cyberinfrastructure Needed to Address the Research Challenge(s) (maximum ~1200 words): Describe any limitations or absence of existing cyberinfrastructure, and/or specific technical advancements in cyberinfrastructure (e.g. advanced computing, data infrastructure, software infrastructure, applications, networking, cybersecurity), that must be addressed to accomplish the identified research challenge(s).

In ICT, storage, networking and processing use a set of node resources and services that overlap to a great degree. However, the node architectures that have evolved to serve these three niches (object storage target, network intermediate node and cluster computing node) have evolved along separate paths, and the system software and high level service architectures built on top of them have served to create highly isolated silos. This isolation is so long-standing and so complete that many ICT experts cannot imagine how they could be viewed as variants of a unified set of abstractions. This conceptual constraint has led to the idea that convergence can only be achieved by high level mechanisms that are users of the services offered by the current storage, networking and processing silos. Again and again such application-level mechanisms have been built but they have failed to achieve deployment scalability.

It is well understood that a standard defined at a lower layer has many advantages over one defined at a high layer of a silo. The reason for this is that any standard tends to restrict heterogeneity at the layer at which it is imposed. Thus a lower layer standard can allow many choices at the layers above it, whereas a high level standard locks in choices that it inherits from the limitations of the silo layers that it relies on. This poses a fundamental problem in designing a common infrastructure for ICT at a very low layer: how can the resources of the intermediate node be modeled in a manner that is general enough to support all necessary services but still be simple, generic and limited? The key lies in finding a unifying abstraction at the lower layer. There may be more than one possible unifying abstraction, but the usual objection is that there is none. So we present our candidate here, not as the only option, but as a proof of feasibility.

The unifying abstraction that we propose for a layer on which storage, networking and processing can all be built is the persistent memory/storage buffer or block. This refers to a segment of memory or storage that has a unique name and which can be 1) allocated, written and read over time, 2) between which data can be transferred and which 3) can be transformed by imperative operations. Our hypothesis is that a service which manages persistent buffers and enables the application of these three classes of service is sufficient to implement a usefully broad class of distributed services and applications. We further hypothesize that if the design of this low layer service follows that hourglass model, with a simple, generic and limited common service layer, that it can achieve deployment scalability and form the basis of a global interoperable infrastructure.

What reason is there to believe that the creation of such a simple, generic and limited service modeling persistent buffers is even possible? A proof of concept exists in the form of the Logistical Networking stack and associated Data Logistics Toolkit, which has been developed over the past 20 years and which has been used in experiments and pre-production infrastructures to implement a wide variety of ICT applications and services. The core buffer management service is the Internet Backplane Protocol (so named because it is implemented as an Internet overlay) augmented with a transformative operations mechanism that has been termed the Network Functional Unit (in analogy to transformative hardware components in processing elements). The DLT has been developed under federal funding (mainly DOE and NSF) and is available as a hardened open source software project installable using a Linux package manager.

To be clear, the research challenge is the creation of a globally scalable, converged ICT infrastructure. Adopting a common model of persistent buffers at the implementation layer below the current layer 3 and the specific example of the Data Logistics Toolkit are proofs of concept and an example candidate. The suggestion is to develop and choose a standard based on community contributions and the consensus of widespread adoption.

But if such a standard is developed, then because it builds on a simple, generic and limited abstraction of basic node resources (i.e., processing, networking, storage), it can support an execution model that does not require that all the data be sent to large scale data centers, and thus is well suited to applications in edge environments. Moreover, since it can use, but does not require, the full services of a traditional operating system as containers do, it can be implemented on an extremely wide range of lighter weight units, including very

small, power-sensitive embedded system class processors, network processors, and FPGAs. Thus, we believe that this design gives our minimalist approach significant advantages over container-based approaches in terms of economic efficiency and deployment scalability.

Consent Statement